Adversary Lower Bound for the Orthogonal Array Problem

Robert Špalek*
spalek@google.com

Abstract

We prove a quantum query lower bound $\Omega(n^{(d+1)/(d+2)})$ for the problem of deciding whether an input string of size n contains a k-tuple which belongs to a fixed orthogonal array on k factors of strength $d \leq k-1$ and index 1, provided that the alphabet size is sufficiently large. Our lower bound is tight when $d = k-1$.

The orthogonal array problem includes the following problems as special cases:

- k-sum problem with $d = k-1$,
- k-distinctness problem with $d = 1$,
- k-pattern problem with $d = 0$,
- $(d-1)$-degree problem with $1 \leq d \leq k-1$,
- unordered search with $d = 0$ and $k = 1$, and
- graph collision with $d = 0$ and $k = 2$.

1 Introduction

1.1 History

One of two main techniques for proving lower bounds on quantum query complexity of Boolean functions is the *adversary method* developed by Ambainis [Amb02, Amb06] and independently by Barnum, Saks, and Szegedy [BSS03] as a generalization of the "hybrid argument" introduced by Bennett, Bernstein, Brassard, and Vazirani [BBBV97] for the Or function.

The adversary bound was strengthened by Høyer, Lee, and Špalek [HLŠ07] to the negative-weight adversary lower bound, which we define in Section 2. This stronger version was proved to be optimal by Reichardt [Rei11], and shown to apply to non-Boolean functions and also to the more general setting of state generation and conversion by Lee, Mittal, Reichardt, Špalek, and Szegedy [LMR$^+$11]. Although the negative-weight adversary lower bound is known to be tight, for a long time it had not been used to prove lower bounds for explicit functions. Vast majority of lower bounds by the adversary method used the old positive-weight version of this method, and the only bounds which utilized the power of negative weights were for functions on a constant number of bits and their compositions, and these bounds were obtained by numeric optimization.

The other main technique for quantum query lower bounds is the *polynomial method* developed by Beals, Buhrman, Cleve, Mosca, and de Wolf [BBC$^+$01]. This method is in general incomparable to the adversary method. Ambainis showed several iterated functions for which the adversary method gives polynomially larger bounds [Amb06]. On the other hand, the polynomial method gives stronger bounds for low-error and zero-error algorithms [BCWZ99].

*Google, Inc.

Another example where the polynomial method used to give stronger bounds than the adversary method is the *element distinctness* function. The input to the function is a string of length n of symbols in an alphabet of size q, i.e., $x = (x_1, \ldots, x_n) \in [q]^n$. We use notation $[q]$ to denote the set $\{1, \ldots, q\}$. The element distinctness function evaluates to 0 if all symbols in the input string are pairwise distinct, and to 1 otherwise. The quantum query complexity of element distinctness is $O(n^{2/3})$ with the algorithm given by Ambainis [Amb07]. Tight lower bounds were given by Aaronson and Shi [AS04], Kutin [Kut05], and Ambainis [Amb05] using the polynomial method.

The positive-weight adversary bound, however, fails for element distinctness. The reason is that this function has 1-certificate complexity 2, and the so-called *certificate complexity barrier* [ŠS06, Zha05] implies that for any function with 1-certificate complexity bounded by a constant, the positive-weight adversary method cannot achieve anything better than $\Omega(\sqrt{n})$. The negative-weight adversary bound is not limited by this barrier [HLŠ07], but showing an explicit adversary lower bound breaking the certificate complexity barrier for this function or, in fact, for any function on more than a constant number of bits was open for a long time.

Belovs and Špalek [BŠ13] were the first to show such an explicit lower bound. They reproved the $\Omega(n^{2/3})$ lower bound for element distinctness using the negative-weight adversary method, and generalized it to the following problem. Let $\mathbb{G}$ be a finite Abelian group, $t \in \mathbb{G}$ its arbitrary element, and k an arbitrary but fixed constant. The *k-sum problem* consists in deciding whether the input string $x_1, \ldots, x_n \in \mathbb{G}$ contains a subset of k elements that sums up to t. This problem was first posed by Childs and Eisenberg [CE05], who noted that it is the hardest problem among all problems with 1-certificate complexity k, because knowledge of any $k-1$ input values doesn't reveal any information about whether that $(k-1)$-tuple can be a part of a 1-certificate or not, and they conjectured that its complexity is $\Omega(n^{k/(k+1)})$. Belovs and Špalek [BŠ13] resolved this conjecture in the positive.

The $\Omega(n^{k/(k+1)})$ lower bound for the k-sum problem is tight thanks to the quantum algorithm based on quantum walks on the Johnson graph [Amb07]. This algorithm was first designed to solve the *k-distinctness problem*. This problem asks for detecting whether the input string $x \in [q]^n$ contains k elements that are all equal. Element distinctness is the same as 2-distinctness. Soon enough it was realized that the same algorithm works for any function with 1-certificate complexity k [CE05], in particular, for the k-sum problem. The quantum query complexity of this algorithm is $O(n^{k/(k+1)})$, and the algorithm is thus optimal for the k-sum problem.

Quantum walk on the Johnson graph is not optimal for the k-distinctness problem when $k > 2$. Belovs and Lee showed a ground-breaking quantum algorithm for a certain promise version of this problem based on learning graphs running in $O(n^{1-2^{k-2}/(2^k-1)}) = o(n^{3/4})$ queries [BL11], which Belovs then improved to an algorithm for full k-distinctiness [Bel12] with the same query complexity. However, none of these two algorithms is time-efficient. Very recently, Belovs and independently Childs, Jeffery, Kothari, and Magniez described two new quantum walk algorithms for 3-distinctness, not based on learning graphs, running in time $\tilde{O}(n^{5/7})$ [Bel13b, CJKM13]. The best known lower bound for the k-distinctness problem is just $\Omega(n^{2/3})$, by a reduction from element distinctness.

1.2 Our result

In this paper, we generalize k-distinctness, k-sum, and several other problems, and express them as special cases of a general family of functions, characterized by orthogonal arrays. Let us define orthogonal arrays first. We use the following notation. For an $x = (x_1, \ldots, x_n) \in X^n$ and $S \subseteq [n]$, let x_S denote the projection of x on S, i.e., the vector $(x_{s_1}, \ldots, x_{s_\ell})$ where $s_1, \ldots, s_\ell$ are the elements of S in the increasing order.

Definition 1 (Orthogonal array [Rao47, HSS99]). Let X be an alphabet. Assume T is a subset of X^k of size $\lambda \cdot |X|^d$ for integers $0 \leq d \leq k-1$ and $\lambda \geq 1$. We say that T is a d-(X, k, λ) *orthogonal array* iff, for every subset of indices $D \subset [k]$ of size d and for every vector $(y_1, \ldots, y_d) \in X^d$, there exist exactly λ strings $(x_1, \ldots, x_k) \in T$ such that $x_D = y$. We call d the *strength*, k the *number of factors*, λ the *index of the array*, and $|X|$ the *alphabet size*. We call T *linear* if X is a finite field, and the elements of T form a subspace of the vector space X^k.

In this paper, we restrict ourselves to orthogonal arrays of index $\lambda = 1$.

Definition 2 (Consistent collection of orthogonal arrays). Assume that each subset S of $[n]$ of size k is equipped with a d-$(X, k, 1)$ orthogonal array T_S. A *collection* $\{T_S\}_S$ *of orthogonal arrays* is called *consistent* iff, for every pair of subsets $S_1, S_2 \subset [n]$ of size k with $|S_1 \cap S_2| \geq d$, their corresponding orthogonal arrays are consistent. We say that T_{S_1} is consistent with T_{S_2} iff, for every $D \subseteq S_1 \cap S_2$ of size d and every vector $(y_1, \ldots, y_d) \in X^d$, the unique vectors $x^1 \in T_{S_1}$ and $x^2 \in T_{S_2}$ satisfying $x^1_D = x^2_D = y$ are consistent on the whole intersection $S_1 \cap S_2$, i.e., $x^1_{S_1 \cap S_2} = x^2_{S_1 \cap S_2}$.

Definition 3 (Orthogonal array problem). Let $\{T_S\}_S$ be a collection of d-(X, k, λ) orthogonal arrays. The d-(X, k, λ) *orthogonal array problem* consists in finding an element of any of the orthogonal arrays in the input string. More precisely, the input $x \in X^n$ evaluates to 1 iff there exists a subset $S \subseteq [n]$ of size k such that $x_S \in T_S$. If the collection is consistent, we call the problem a *consistent orthogonal array problem*.

The orthogonal array problem was first defined by Belovs and Špalek [BŠ13] as a convenient tool to prove a tight lower bound for the k-sum problem, and it was also used by Belovs and Rosmanis [BR12] to prove a lower bound on the quantum query complexity of certificate structures. Both these papers only use a special case of orthogonal arrays with strength $k-1$, whereas we allow for any strength $d \leq k-1$.

Consider the following orthogonal array problems. The first three examples have been widely studied before. The last two examples are new, at least in the context of this paper.

Example 1 (k-distinctness problem [Amb07, Bel12]). Let X be any alphabet. $T = \{x^k : x \in X\}$ is a 1-$(X, k, 1)$ orthogonal array. A collection of these arrays is consistent.

Example 2 (k-sum problem [CE05, BŠ13]). Let $\mathbb{G}$ be an Abelian group and $t \in \mathbb{G}$. $T = \{(x_1, \ldots, x_k) \in \mathbb{G}^k : \sum_{i=1}^k x_i = t\}$ is a $(k-1)$-$(\mathbb{G}, k, 1)$ orthogonal array. A collection of these arrays is consistent.

Example 3 (Unordered search [BBBV97, Gro97]). Let X be any alphabet and $x \in X$. $T = \{x\}$ is a 0-$(X, 1, 1)$ orthogonal array. Unordered search is equal to the 1-sum problem.

Example 4 (k-pattern problem). Let X be any alphabet. For each k-tuple S, fix a string $y^S \in X^k$. $T_S = \{y^S\}$ is a 0-$(X, k, 1)$ orthogonal array.

If the collection $\{T_S\}_S$ of the orthogonal arrays is consistent, then the k-pattern problem is equivalent to k unordered searches without replacement, because there exists a unique vector $y \in X^n$ such that $y^S = y_S$. If the collection is inconsistent, then the k-pattern problem is more general than unordered search. For example, the *graph collision* problem [MSS07] is a special case of the 2-pattern problem. See our open problems for a more detailed discussion.

Example 5 (d-degree problem). Let $\mathbb{F}$ be a finite field and $0 \leq d \leq k-2$. For each k-tuple S, let $T_S = \{x_S \in \mathbb{F}^k : \exists \alpha_0, \ldots \alpha_d \in \mathbb{F} : \forall s \in S : x_s = \sum_{i=0}^d \alpha_i s^i\}$. T_S is a linear $(d+1)$-$(\mathbb{F}, k, 1)$ orthogonal array.

A collection of these orthogonal arrays is consistent thanks to the way we consistently use the indices $s \in S$ as the points at which the polynomials are evaluated. Had we, for example, instead sorted the elements of S in an increasing order, indexed them by $[k]$, and defined the

polynomial at these points, we would have obtained a different collection of $(d+1)$-$(\mathbb{F}, k, 1)$ orthogonal arrays, which is not consistent.

k-distinctness and k-sum represent two extreme examples of orthogonal array problems, differing by their strength, and the d-orthogonal array problem naturally interpolates between them. Given that the quantum query complexity of the k-sum problem is known and the complexity of k-distinctness is open, it is natural to ask how large lower bound can one prove for the d-orthogonal array problem, as a function of d. We address this question and prove the following result.

Theorem 4 (Main result)**.** *For a fixed k and $0 \leq d \leq k-1$, an alphabet X, and any collection of d-$(X, k, 1)$ orthogonal arrays T_S, the quantum query complexity of the d-$(X, k, 1)$ orthogonal array problem is $\Omega(n^{(d+1)/(d+2)})$ provided that $|X| \geq n^{k/(k-d)}$. The constant behind the big-Omega depends on k and d, but not on n, $|X|$, or the choice of T_S. The collection of orthogonal arrays may or may not be consistent.*

The proofs in our paper are straightforward extensions of the corresponding proofs of the quantum query lower bound for the k-sum problem [BŠ13].

Our lower bound for k-distinctness is direct, meaning that it doesn't use reduction from element distinctness, and it gives the same bound $\Omega(n^{2/3})$. The lower bound for the d-orthogonal array problem grows with growing d until it reaches its maximal value $\Omega(n^{k/(k+1)})$ for the k-sum problem, where the bound is optimal. We don't know whether our bound is optimal for any $d < k-1$.

We conjecture that any consistent d-$(X, k, 1)$ orthogonal array problem can be solved in $o(n^{k/(k+1)})$ quantum queries when $d < k-1$, using learning graphs like in [Bel12]. That includes the $(d-1)$-degree problem. Finding such an algorithm is one of our open problems.

2 Adversary Lower Bound

In this paper we are interested in the quantum query complexity of the d-$([q], k, 1)$ orthogonal array problem. For the definitions and main properties of quantum query complexity refer to, e.g., Ref. [BW02]. For the history, definitions, and relationships between various quantum query lower-bound methods refer to, e.g., Ref. [HŠ05]. For the purposes of our paper, it is enough to define the adversary bound, which we do in this section.

We use the formulation from Ref. [BŠ13]. Compared to the original formulation of the negative-weight adversary bound [HLŠ07], this formulation is different in two aspects. First, in order to simplify the notation, we call an adversary matrix a matrix with rows labeled by positive inputs, and columns by negative inputs. It is a quarter of the original adversary matrix that completely specifies the latter. Second, due to technical reasons, we allow several rows to be labeled by the same positive input. All this is captured by the following definition and theorem.

Definition 5. Let f be a function $f\colon \mathcal{D} \to \{0,1\}$ with domain $\mathcal{D} \subseteq [q]^n$. Let $\widetilde{\mathcal{D}}$ be a set of pairs (x, a) with the property that the first element of each pair belongs to $\mathcal{D}$, and $\widetilde{\mathcal{D}}_i = \{(x, a) \in \widetilde{\mathcal{D}} : f(x) = i\}$ for $i \in \{0, 1\}$. An *adversary matrix* for the function f is a non-zero real $\widetilde{\mathcal{D}}_1 \times \widetilde{\mathcal{D}}_0$ matrix Γ. For an $i \in [n]$, let Δ_i denote the $\widetilde{\mathcal{D}}_1 \times \widetilde{\mathcal{D}}_0$ matrix defined by

$$\Delta_i[\![(x, a), (y, b)]\!] = \begin{cases} 0, & x_i = y_i; \\ 1, & \text{otherwise.} \end{cases}$$

Theorem 6 (Adversary bound [HLŠ07, BŠ13]). *In the notation of Definition 5,* $Q_2(f) = \Omega(\mathrm{Adv}^{\pm}(f))$*, where*

$$\mathrm{Adv}^{\pm}(f) = \max_{\Gamma} \frac{\|\Gamma\|}{\max_{i \in n} \|\Gamma \circ \Delta_i\|} \tag{1}$$

where the maximization is over all adversary matrices for f*,* $\|\cdot\|$ *is the spectral norm, and* $Q_2(f)$ *is the quantum query complexity of* f*.*

3 Proof

In this section we prove Theorem 4 using the adversary lower bound, Theorem 6. The idea of our construction is to embed the adversary matrix Γ into a slightly larger matrix $\widetilde{\Gamma}$ with additional columns. Then $\Gamma \circ \Delta_i$ is a sub-matrix of $\widetilde{\Gamma} \circ \Delta_i$, hence, $\|\Gamma \circ \Delta_i\| \le \|\widetilde{\Gamma} \circ \Delta_i\|$. (In this section we use Δ_i to denote all matrices defined like in Definition 5, with the size and the labels of the rows and columns clear from the context.) It remains to prove that $\|\widetilde{\Gamma}\|$ is large, and that $\|\Gamma\|$ is not much smaller than $\|\widetilde{\Gamma}\|$.

The proof is organized as follows. In Section 3.1 we define $\widetilde{\Gamma}$ depending on certain parameters α_m, in Section 3.2 we analyze its norm, in Sections 3.3 and 3.4 we calculate $\|\widetilde{\Gamma} \circ \Delta_i\|$, in Section 3.5 we optimize α_m, and, finally, in Section 3.6 we prove that the norm of the true adversary matrix Γ is not much smaller than the norm of $\widetilde{\Gamma}$.

3.1 Adversary matrix

Matrix $\widetilde{\Gamma}$ consists of $\binom{n}{k}$ matrices $\widetilde{G}_{s_1,\ldots,s_k}$ stacked one on another for all possible choices of subset $S = \{s_1, \ldots, s_k\} \subset [n]$:

$$\widetilde{\Gamma} = \begin{pmatrix} \widetilde{G}_{1,2,\ldots,k} \\ \widetilde{G}_{1,2,\ldots,k-1,k+1} \\ \cdots \\ \widetilde{G}_{n-k+1,n-k+2,\ldots,n} \end{pmatrix} . \tag{2}$$

Each $\widetilde{G}_S$ is a $q^{n-k+d} \times q^n$ matrix with rows indexed by inputs $(x_1, \ldots, x_n) \in [q]^n$ such that $x_S \in T_S$, and columns indexed by all possible inputs $(y_1, \ldots, y_n) \in [q]^n$.

We say that a column with index y is *invalid* if $y_S \in T_S$ for some $S \subseteq [n]$. After removing all invalid columns, $\widetilde{G}_S$ will represent the part of Γ with the rows indexed by the inputs having an element of the orthogonal array on S. Note that some positive inputs appear more than once in Γ. More specifically, an input x appears as many times as many elements of the orthogonal arrays it contains.

This construction may seem faulty, because there are elements of $[q]^n$ that are used as labels of both rows and columns in $\widetilde{\Gamma}$, and hence, it is trivial to construct a matrix $\widetilde{\Gamma}$ such that the value in (1) is arbitrarily large. However, we design $\widetilde{\Gamma}$ in a specifically restrictive way so that it still is a good adversary matrix after the invalid columns are removed.

Let J_q be the $q \times q$ all-ones matrix. Assume $e_0, \ldots, e_{q-1}$ is an orthonormal eigenbasis of J_q with $e_0 = 1/\sqrt{q} \cdot (1, \ldots, 1)$ being the eigenvalue q eigenvector. Consider the vectors of the following form:

$$v = e_{v_1} \otimes e_{v_2} \otimes \cdots \otimes e_{v_n}, \tag{3}$$

where $v_i \in \{0, \ldots, q-1\}$. These are eigenvectors of the Hamming Association Scheme on $[q]^n$. For a vector v from (3), the *weight* $|v|$ is defined as the number of non-zero entries in $(v_1, \ldots, v_n)$. Let $E_k^{(n)}$, for $k = 0, \ldots, n$, be the orthogonal projector onto the space spanned by the vectors

from (3) having weight k. These are the projectors on the eigenspaces of the association scheme. Let us denote $E_i = E_i^{(1)}$ for $i = 0, 1$. These are $q \times q$ matrices. All entries of E_0 are equal to $1/q$, and the entries of $E_1 = I - E_0$ are given by

$$E_1[\![x, y]\!] = \begin{cases} 1 - 1/q, & x = y; \\ -1/q, & x \neq y. \end{cases}$$

Elements of S in $\widetilde{G}_S$ should be treated differently from the remaining elements. For them, we define a $q^d \times q^k$ matrix F_S. It has rows labelled by the elements of T_S and columns by the elements of $[q]^k$, and is defined as follows.

Definition 7. Let

$$E_{\leq d}^{(k)} = \sum_{i=0}^{d} E_i^{(k)} = \sum_{\substack{u = e_{u_1} \otimes \cdots \otimes e_{u_k} \\ |u| \leq d}} uu^*$$

be the projector onto the subspace spanned by the vectors of weight at most d. Let F_S be $q^{(k-d)/2}$ times the sub-matrix of $E_{\leq d}^{(k)}$ consisting of only the rows from T_S.

Finally, we define $\widetilde{\Gamma}$ as in (2) with $\widetilde{G}_S$ defined by

$$\widetilde{G}_S = \sum_{m=0}^{n-k} \alpha_m F_S \otimes E_m^{(n-k)} \ , \tag{4}$$

where F_S acts on the elements in S and E_m acts on the remaining $n-k$ elements. The coefficients α_m will be specified later.

3.2 Norm of $\widetilde{\Gamma}$

Lemma 8. *Let $\widetilde{\Gamma}$ be like in (2) with $\widetilde{G}_S$ defined as in (4). Then*

(a) $\|\widetilde{\Gamma}\| = \Omega(\alpha_0 n^{k/2})$,

(b) $\|\widetilde{\Gamma}\| = O(\max_m \alpha_m n^{k/2})$.

Proof. Fix a subset S and denote $T = T_S$ and $F = F_S$. Recall that $E_{\leq d}^{(k)}$ is the sum of uu^* over all $u = e_{u_1} \otimes \cdots \otimes e_{u_k}$ with at least $k - d$ elements u_j equal to 0, and F is the restriction of $E_{\leq d}^{(k)}$ to the rows in T.

For $u = e_{u_1} \otimes \cdots \otimes e_{u_k}$ and $L \subset [k]$ of size $|L| = k - d$ such that $u_L = e_0^{\otimes(k-d)}$, let u^L denote the $q^{(k-d)/2}$ multiple of u restricted to the elements in T. The reason for the superscript is that we consider the following process of obtaining u^L: we treat T as $[q]^d$ by erasing the elements indexed by L in any string of T, then u^L coincides on this set with u with the L-terms removed.

In this notation, the contribution from uu^* to F equals $u^{L_u}u^*$, where L_u is any set of $k - d$ positions in u containing e_0. In general, we do not know how the u^L vectors relate for different L. However, we know that, for a fixed L, they are all orthogonal; and for any L, $(e_0^{\otimes k})^L$ is the vector $1/\sqrt{q^d} \cdot (1, \ldots, 1)$.

Let us start with proving (a). We estimate $\|\widetilde{\Gamma}\|$ from below by $w^*\widetilde{\Gamma}w'$, where w and w' are unit vectors with all elements equal. In other words, $\|\widetilde{\Gamma}\|$ is at least the sum of all its entries divided by $\sqrt{\binom{n}{k} q^{2n+d-k}}$. In order to estimate the sum of the entries of $\widetilde{\Gamma}$, we rewrite (4) as

$$\widetilde{G}_S = \alpha_0 e_0^{\otimes(n+d-k)} (e_0^{\otimes n})^* + \sum_{u,v} \alpha_{|v|} (u^{L_u} \otimes v)(u \otimes v)^* \ , \tag{5}$$

where the summation is over all u and v such that at least one of them contains an element different from e_0. The sum of all entries in the first term of (5) is $\alpha_0 q^{n+(d-k)/2}$. The sum of each column in each of $(u^{L_u} \otimes v)(u \otimes v)^*$ is zero because at least one of u^{L_u} or v sums up to zero. By summing over all $\binom{n}{k}$ choices of S, we get that $\|\widetilde{\Gamma}\| \geq \alpha_0 \sqrt{\binom{n}{k}} = \Omega(\alpha_0 n^{k/2})$.

In order to prove (b), express F_S as $\sum_{L \subset [k]:|L|=k-d} F_S^L$ with $F_S^L = \sum_{u \in U_L} u^L u^*$. Here $\{U_L\}$ is an arbitrary decomposition of all u such that U_L contains only u with e_0 in the L-positions. Define $\widetilde{G}_S^L$ as in (4) with F_S replaced by F_S^L, and $\widetilde{\Gamma}^L$ as in (2) with $\widetilde{G}_S$ replaced by $\widetilde{G}_S^L$.

Since all u^L vectors are orthogonal for a fixed L, we get that

$$(\widetilde{G}^L)^* \widetilde{G}^L = \sum_{u \in U_L, v} \alpha_{|v|}^2 (u \otimes v)(u \otimes v)^*,$$

thus $\|(\widetilde{G}^L)^* \widetilde{G}^L\| = \max_m \alpha_m^2$. By the triangle inequality,

$$\|\widetilde{\Gamma}^L\|^2 = \left\|(\widetilde{\Gamma}^L)^* \widetilde{\Gamma}^L\right\| = \left\|\sum_S (\widetilde{G}_S^L)^* \widetilde{G}_S^L\right\| \leq \binom{n}{k} \max_m \alpha_m^2.$$

Since $\widetilde{\Gamma} = \sum_{L \subset [k]:|L|=k-d} \widetilde{\Gamma}^L$ and $\binom{k}{k-d} = O(1)$, another application of the triangle inequality finishes the proof of (b). □

3.3 Action of Δ_1

The adversary matrix is symmetric in all input variables and hence it suffices to only consider the entry-wise multiplication by Δ_1. Precise calculation of $\|\widetilde{\Gamma} \circ \Delta_1\|$ is very tedious, but one can get an asymptotically tight bound using the following trick. Instead of computing $\widetilde{\Gamma} \circ \Delta_1$ directly, we arbitrarily map $\widetilde{\Gamma} \overset{\Delta_1}{\longmapsto} \widetilde{\Gamma}_1$ such that $\widetilde{\Gamma}_1 \circ \Delta_1 = \widetilde{\Gamma} \circ \Delta_1$, and use the inequality $\|\widetilde{\Gamma}_1 \circ \Delta_1\| \leq 2\|\widetilde{\Gamma}_1\|$ that holds thanks to $\gamma_2(\Delta_1) \leq 2$ [LMR$^+$11]. In other words, we change arbitrarily the entries with $x_1 = y_1$. We use the mapping

$$E_0 \overset{\Delta_1}{\longmapsto} E_0 \ , \qquad E_1 \overset{\Delta_1}{\longmapsto} -E_0 \ . \tag{6}$$

The projector $E_{\leq d}^{(k)}$ is mapped by Δ_1 as

$$E_{\leq d}^{(k)} \ \overset{\Delta_1}{\longmapsto} \ E_0 \otimes E_d^{(k-1)} \ . \tag{7}$$

It follows that

$$F \ \overset{\Delta_1}{\longmapsto} \ F_1 = \sum_{\substack{u = e_{u_1} \otimes \cdots \otimes e_{u_k} \\ u_1 = 0, |u| = d}} u^{L_u} u^* \ , \tag{8}$$

where u^{L_u} is defined like in the proof of Lemma 8. For a subset $L \subset [k]$ of size $|L| = k - d$ that contains $1 \in L$,

$$F^L \ \overset{\Delta_1}{\longmapsto} \ (F^L)_1 = \sum_{\substack{u \in U_L \\ |u| = d}} u^L u^* \ , \tag{9}$$

where F^L and U_L is defined like in the proof of Lemma 8(b).

3.4 Norm of $\widetilde{\Gamma}_1$

Lemma 9. *Let $\widetilde{\Gamma}$ be like in (2) with $\widetilde{G}_S$ defined as in (4), and map $\widetilde{\Gamma} \stackrel{\Delta_1}{\longmapsto} \widetilde{\Gamma}_1$, $\widetilde{G}_S \stackrel{\Delta_1}{\longmapsto} (\widetilde{G}_S)_1$, and $F_S \stackrel{\Delta_1}{\longmapsto} (F_S)_1$ using (6) and (8). Then*

$$\|\widetilde{\Gamma}_1\| = O\left(\max_m \left(\max(\alpha_m m^{d/2} n^{(k-1-d)/2}, (\alpha_m - \alpha_{m+1}) n^{k/2})\right)\right).$$

Proof. In order to prove the upper bound, we express $\widetilde{\Gamma}_1 = \sum_L \widetilde{\Gamma}_1^L$, $(\widetilde{G}_S)_1 = \sum_L (\widetilde{G}_S^L)_1$, and $(F_S)_1 = \sum_{L\subset[k]:|L|=k-d} (F_S^L)_1$, like in the proof of Lemma 8(b), and upper-bound each $\|\widetilde{\Gamma}_1^L\|$ separately. Note that even though L is a subset of $[k]$ and not S, we can still use L to select a subset of elements of S if each S is ordered in the ascending order. $S_L = \{s_i : i \in L\}$ for $S = \{s_1, \ldots, s_k\}$.

We have $\|\widetilde{\Gamma}_1^L\|^2 = \|(\widetilde{\Gamma}_1^L)^* \widetilde{\Gamma}_1^L\| = \|\sum_S (\widetilde{G}_S^L)_1^* (\widetilde{G}_S^L)_1\|$. Decompose the set of all possible k-tuples of indices into $\mathcal{S}_1 \cup \mathcal{S}_2$, where $\mathcal{S}_1$ are k-tuples containing 1 and $\mathcal{S}_2$ are k-tuples that don't contain 1. We upper-bound the contribution of $\mathcal{S}_1$ to $\|\widetilde{\Gamma}_1^L\|^2$ by $\max_m \alpha_m^2 \binom{m+d}{d}\binom{n-m-d-1}{k-1-d}$ and the contribution of $\mathcal{S}_2$ by $\max_m(\alpha_m - \alpha_{m+1})^2 \binom{n-1}{k}$, and apply the triangle inequality.

Let $v = e_{v_1} \otimes \cdots \otimes e_{v_n}$ with $|v| = m + d$, and let $S \in \mathcal{S}_1$. Then, by (9),

$$(\widetilde{G}_S^L)_1 v = \begin{cases} \alpha_m v^{S_L}, & v_1 = 0,\ |v_S| = d, \text{ and } |v_{S_L}| = 0 \\ 0, & \text{otherwise.} \end{cases}$$

Here $v_S = \bigotimes_{s\in S} e_{v_s}$ and $v^{S_L} = q^{(k-d)/2} \bigotimes_{i\in[n]-S_L} e_{v_i}$.

For different v, these are orthogonal vectors, and hence v is an eigenvector of $(\widetilde{G}_S^L)_1^* (\widetilde{G}_S^L)_1$ of eigenvalue α_m^2 if $v_1 = 0$, $|v_S| = d$, and $|v_{S_L}| = 0$, and of eigenvalue 0 otherwise. For every v with $v_1 = 0$ and $|v| = m + d$, there are $\binom{m+d}{d}\binom{n-m-d-1}{k-1-d}$ sets $S \in \mathcal{S}_1$ such that $|v_S| = d$, and hence at most as many sets $S \in \mathcal{S}_1$ such that $(\widetilde{G}_S^L)_1 v \neq 0$. We apply the triangle inequality, and conclude that the contribution of $\mathcal{S}_1$ is as claimed.

Now consider an $S \in \mathcal{S}_2$, that means $1 \notin S$.

$$\begin{aligned} \widetilde{G}_S^L &= \sum_{m=0}^{n-k} \alpha_m F_S^L \otimes E_m^{(n-k)} \\ &= \sum_{m=0}^{n-k} \alpha_m F_S^L \otimes (E_0 \otimes E_m^{(n-k-1)} + E_1 \otimes E_{m-1}^{(n-k-1)}) \\ &\stackrel{\Delta_1}{\longmapsto} \sum_{m=0}^{n-k} \alpha_m F_S^L \otimes E_0 \otimes (E_m^{(n-k-1)} - E_{m-1}^{(n-k-1)}) \\ = (\widetilde{G}_S^L)_1 &= \sum_{m=0}^{n-k} (\alpha_m - \alpha_{m+1}) F_S^L \otimes E_0 \otimes E_m^{(n-k-1)} \ . \end{aligned}$$

Therefore $(\widetilde{G}_S^L)_1$ is of the same form as $\widetilde{G}_S^L$, but with coefficients $(\alpha_m - \alpha_{m+1})$ instead of α_m and on one dimension less. We get the required estimate from Lemma 8(b).

There are $\binom{k}{k-d}$ sets $L \subset [k]$ of size $|L| = k - d$. Since $k, d = O(1)$, one more application of the triangle equality gets the claimed bound. □

3.5 Optimization of α_m

To maximize the adversary bound, we maximize $\|\widetilde{\Gamma}\|$ while keeping $\|\widetilde{\Gamma}_1\| = O(1)$. That means, we choose the coefficients $\{\alpha_m\}$ to maximize $\alpha_0 n^{k/2}$ (Lemma 8) so that, for every m, $\alpha_m \leq m^{-d/2} n^{(d+1-k)/2}$ and $\alpha_m \leq \alpha_{m+1} + n^{-k/2}$ (Lemma 9).

For every $r \in [n]$, $\alpha_0 \leq \alpha_r + r n^{-k/2} \leq r^{-d/2} n^{(d+1-k)/2} + r n^{-k/2}$. The expression on the right-hand side achieves its minimum, up to a constant, $\alpha_0 = 2\ n^{(d+1)/(d+2)-k/2}$ for $r = n^{(d+1)/(d+2)}$. This corresponds to the following solution:

$$\alpha_m = \max\left\{2 - \frac{m}{n^{(d+1)/(d+2)}}, 0\right\} n^{(d+1)/(d+2)-k/2} \tag{10}$$

With this choice of α_m, $\|\widetilde{\Gamma}\| = \Omega(\alpha_0 n^{k/2}) = \Omega(n^{(d+1)/(d+2)})$.

3.6 Constructing Γ from $\widetilde{\Gamma}$

The matrix $\widetilde{\Gamma}$ gives us the desired ratio of norms of $\widetilde{\Gamma}$ and $\widetilde{\Gamma} \circ \Delta_i$. Unfortunately, $\widetilde{\Gamma}$ cannot directly be used as an adversary matrix, because it contains invalid columns y with $f(y) = 1$, that is, y that contain an element of the orthogonal array on $S \subset [n] : |S| = k$, i.e., $y_S \in T_S$. We show that after removing the invalid columns the adversary matrix Γ is still good enough.

Lemma 10. *Let Γ be the sub-matrix of $\widetilde{\Gamma}$ with the invalid columns removed. Then $\|\Gamma \circ \Delta_1\| \leq \|\widetilde{\Gamma} \circ \Delta_1\|$, and $\|\Gamma\|$ is still $\Omega(\alpha_0 n^{k/2})$ when $q \geq n^{k/(k-d)}$.*

Proof. We estimate $\|\Gamma\|$ from below by $w^* \Gamma w'$ using unit vectors w, w' with all elements equal. Recall Equation (5):

$$\widetilde{G}_S = \alpha_0 e_0^{\otimes(n+d-k)} (e_0^{\otimes n})^* + \sum_{u,v} \alpha_{|v|} (u^{L_u} \otimes v)(u \otimes v)^* \ ,$$

where the summation is over all u and v such that at least one of them contains an element different from e_0. The sum of each column in each of $(u^{L_u} \otimes v)(u \otimes v)^*$ is still zero because at least one of u^{L_u} or v sums up to zero. Therefore the contribution of the sum is zero regardless of which columns have been removed.

By summing over all $\binom{n}{k}$ choices of S, we get

$$\|\Gamma\| \geq w^* \Gamma w' = \sqrt{\binom{n}{k}} \alpha_0 \ (e_0^{\otimes n})_V^* w' \ ,$$

where e_V denotes the sub-vector of e restricted to V, and V is the set of valid columns. Since both e_0 and w' are unit vectors with all elements equal, and w' is supported on V, $(e_0^{\otimes n})_V^* w' = \sqrt{|V|/q^n}$.

Let us estimate the fraction of valid columns. The probability that a uniformly random input $y \in [q]^n$ contains an orthogonal array at any given k-tuple S is q^{d-k}. By the union bound, the probability that there exists such S is at most $\binom{n}{k} q^{d-k}$. Therefore the probability that a random column is valid is $|V|/q^n \geq 1 - \binom{n}{k} q^{d-k}$, which is $\Omega(1)$ when $q \geq n^{k/(k-d)}$. □

Thus, with the choice of α_m from (10), we have $\mathrm{Adv}^{\pm}(f) = \Omega(\alpha_0 n^{k/2}) = \Omega(n^{(d+1)/(d+2)})$. This finishes the proof of Theorem 4.

4 Open problems

- Our lower bound $\Omega(n^{(d+1)/(d+2)})$ for the d-$(X,k,1)$ orthogonal array problem is only known to be optimal when the strength $d = k-1$. This variant corresponds to the k-sum problem [BŠ13], for which one can prove a matching $O(n^{k/(k+1)})$ upper bound by quantum search on the Johnson graph [Amb07]. For the k-distinctness problem, which lies at the other end of the spectrum with the strength $d = 1$, there is a quantum algorithm running in $O(n^{1-2^{k-2}/(2^k-1)}) = o(n^{3/4})$ queries [Bel12], which is polynomially faster for $k \geq 3$. Can one close the gap, say, in the simplest case $d = 1$ and $k = 3$, whose complexity lies between $\Omega(n^{2/3})$ and $O(n^{5/7})$?

  Our lower bound only depends on d but not on k, as long as $k = O(1)$. This seems unlikely to be optimal. Can one strengthen the lower bound for larger k?

- Consider the k-pattern problem, i.e., the 0-$(X,k,1)$ orthogonal array problem. If the patterns are consistent, then the problem is equivalent to k repeated unordered searches without replacement, and its complexity is $\Theta(\sqrt{n})$. If the patterns are inconsistent, then our lower bound stays $\Omega(\sqrt{n})$, but the best known upper bound is just $O(n^{k/(k+1)})$.

  The inconsistent k-pattern problem includes several interesting problems as special cases. For example, graph collision [MSS07] is a 2-pattern problem and finding an ℓ-clique is an $\binom{\ell}{2}$-pattern problem [Bel13a]. Given a fixed graph (V,E) on n vertices and an n-bit input x, the *graph collision problem* is to decide whether there exists an edge $\{i,j\} \in E$ with $x_i = x_j = 1$. Given a fixed vertex set V, and edges E specified by an input black-box, the *ℓ-clique problem* is to decide whether the graph (V,E) contains a clique of size ℓ. Both these problems look solely for input variables labeled by 1, and the hardness of the problem comes from the fact that not every subset of input variables is admissible. The patterns specified for non-edges resp. non-cliques of the graphs are labeled by a dummy symbol that is not a part of the input alphabet.

  Our lower bound works regardless of whether the orthogonal arrays are consistent or not, which means that it might not be strong enough for inconsistent orthogonal arrays. Can one prove an $\omega(\sqrt{n})$ lower bound for the inconsistent k-pattern problem? Proving this would be a good step towards proving an $\omega(\sqrt{n})$ lower bound for graph collision.

  It is conceivable that the query complexity of the k-pattern problem can be anything between $\Omega(\sqrt{n})$ and $O(n^{k/(k+1)})$, depending on the combinatorial structure of the collection of patterns. For a consistent collection, we get $\Theta(\sqrt{n})$, and the more "inconsistent" the orthogonal arrays are the larger the lower bound might be. Can one lower-bound the query complexity of the inconsistent k-pattern problem in terms of some positive semidefinite program simpler than the full negative-weight adversary bound? Using duality of semidefinite programming, can one then find a matching quantum algorithm, like in Ref. [Rei11]?

- It is conceivable that the learning graph for k-distinctness [Bel12] can be "interpolated" with the learning graph for the k-sum problem, and solve the consistent d-$(X,k,1)$ orthogonal array problem. (Essentially, one would load the first d elements normally, and the remaining $k-d$ elements with only partial uncovering of loaded elements.) Unfortunately, there are many subtle details in the analysis of the learning graph for k-distinctness, which makes the task of generalizing it difficult. If one addresses all issues, what would the complexity of the learning graph for the consistent d-$(X,k,1)$ orthogonal array problem be, as a function of d? It will probably not match our lower bound, since there is currently a gap even for k-distinctness (with $d = 1$), but can one at least design a quantum algorithm

that for a fixed $d > 1$ runs faster than $n^{1-\Omega(1)}$ for all k, i.e., whose complexity doesn't approach $\Omega(n^{1-o(1)})$ when k grows?

The $o(n^{3/4})$-complexity learning graph for k-distinctness [Bel12] can be cast as a learning graph for the consistent 1-$(X, k, 1)$ orthogonal array problem. The learning graph crucially depends on the consistency of the orthogonal sets. Can one generalize this learning graph to not require consistent orthogonal sets? This is likely to be hard, witnessed by the rich combinatorial structure of the inconsistent k-pattern problem.

- Belovs and Rosmanis have recently generalized the $\Omega(n^{k/(k+1)})$ lower bound for the k-sum problem [BŠ13] to a more general framework of certificate structures [BR12]. Roughly speaking, they show strong lower bounds for the learning graph complexity of several common *certificate structures* (for example, $\tilde{\Omega}(n^{9/7})$ for triangle finding) and then they show that for each certificate structure there exists a black-box function with that certificate structure whose query complexity satisfies the same lower bound. Their functions are based on orthogonal arrays of strength $k-1$ when the 1-certificate size is k. Their collections of orthogonal arrays are consistent, because any collection of $(k-1)$-$(X, k, 1)$ orthogonal sets is necessarily consistent. In the case of triangle finding, their method gives a nearly tight lower bound for the *triangle sum problem*. Can their method be combined with our result to obtain nontrivial quantum query lower bounds for functions based on orthogonal arrays of smaller strengths?

- Our technique relies crucially on the $n^{k/(k-d)}$ lower bound on the alphabet size. Can one relax this bound? This will probably require an entirely new design of the adversary matrix.

- We have only proved a lower bound for the d-(X, k, λ) orthogonal array problem with index $\lambda = 1$. Extending our proof to larger λ seems straightforward. Is there a natural problem with $\lambda > 1$ for which one can prove a nontrivial lower bound?

Acknowledgments

We thank Aleksandrs Belovs and Ansis Rosmanis for valuable discussions.

Most of our proofs are very similar to the corresponding proofs for the quantum query lower bound of the k-sum problem [BŠ13]. We thank Aleksandrs Belovs for agreeing to use their proofs as the basis of our paper.

References

[Amb02] A. Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. Earlier version in STOC'00.

[Amb05] A. Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1:37–46, 2005.

[Amb06] A. Ambainis. Polynomial degree vs. quantum query complexity. *Journal of Computer and System Sciences*, 72(2):220–238, 2006. Earlier version in FOCS'03.

[Amb07] A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. Earlier version in FOCS'04.

[AS04] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004.

[BBBV97] H. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997.

[BBC+01] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. Earlier version in FOCS'98.

[BCWZ99] H. Buhrman, R. Cleve, R. de Wolf, and Ch. Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proc. of 40th IEEE FOCS*, pages 358–368, 1999.

[Bel12] A. Belovs. Learning-graph-based quantum algorithm for k-distinctness. arXiv:1205.1534 [quant-ph], 2012.

[Bel13a] A. Belovs. Personal communication, March 2013.

[Bel13b] A. Belovs. Quantum walks and electric networks. arXiv:1302.3143 [quant-ph], 2013.

[BL11] A. Belovs and T. Lee. Quantum algorithm for k-distinctness with prior knowledge on the input. arXiv:1108.3022 [quant-ph], 2011.

[BR12] A. Belovs and A. Rosmanis. On the power of non-adaptive learning graphs. arXiv:1210.3279 [quant-ph], 2012.

[BŠ13] A. Belovs and R. Špalek. Adversary lower bound for the k-sum problem. In *Proc. of 4th ACM ITCS*, pages 323–328, 2013.

[BSS03] H. Barnum, M. Saks, and M. Szegedy. Quantum decision trees and semidefinite programming. In *Proc. of 18th IEEE Complexity*, pages 179–193, 2003.

[BW02] H. Buhrman and R. de Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[CE05] A.M. Childs and J.M. Eisenberg. Quantum algorithms for subset finding. *Quantum Information & Computation*, 5(7):593–604, 2005.

[CJKM13] A. M. Childs, S. Jeffery, R. Kothari, and F. Magniez. A time-efficient quantum walk for 3-distinctness using nested updates. arXiv:1302.7316 [quant-ph], 2013.

[Gro97] L. K. Grover. Quantum mechanics helps in searching for a needle in a haystack. *Physical Review Letters*, 79(2):325–328, 1997. Earlier version in STOC'96.

[HLŠ07] P. Høyer, T. Lee, and R. Špalek. Negative weights make adversaries stronger. In *Proc. of 39th ACM STOC*, pages 526–535, 2007.

[HŠ05] P. Høyer and R. Špalek. Lower bounds on quantum query complexity. *EATCS Bulletin*, 87:78–103, October, 2005.

[HSS99] A. S. Hedayat, N. J. A. Sloane, and J. Stufken. *Orthogonal arrays: theory and applications*. Springer, 1999.

[Kut05] S. Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing*, 1:29–36, 2005.

[LMR+11] T. Lee, R. Mittal, B. W. Reichardt, R. Špalek, and M. Szegedy. Quantum query complexity of state conversion. In *Proc. of 52nd IEEE FOCS*, pages 344–353, 2011.

[MSS07] F. Magniez, M. Santha, and M. Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007. Earlier version in SODA'05.

[Rao47] C. R. Rao. Factorial experiments derivable from combinatorial arrangements of arrays. *Supplement to the Journal of the Royal Statistical Society*, 9(1):128–139, 1947.

[Rei11] Ben W. Reichardt. Reflections for quantum query algorithms. In *Proc. of 22nd ACM-SIAM SODA*, pages 560–569, 2011.

[ŠS06] R. Špalek and M. Szegedy. All quantum adversary methods are equivalent. *Theory of Computing*, 2(1):1–18, 2006. Earlier version in ICALP'05.

[Zha05] S. Zhang. On the power of Ambainis's lower bounds. *Theoretical Computer Science*, 339(2–3):241–256, 2005. Earlier version in ICALP'04.